



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/846,067	04/30/2001	Rahul Sharma	SUNMP007	4516

25920 7590 08/22/2005

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER

DAO, THUY CHAN

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 08/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/846,067

Applicant(s)

SHARMA ET AL.

Examiner

Thuy Dao

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 31 May 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3,4,6-8 and 10-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3,4,6-8 and 10-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 April 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on May 31, 2005 has been entered.

Response to Amendments

2. Claims 8 and 15 have been amended. Claims 2, 5, and 9 are cancelled. Claims 1, 8, and 15 are independent claims.

Priority

3. The priority date considered for this application is April 30, 2001.

Information Disclosure Statement

4. The Office acknowledges receipts of the Information Disclosure Statement filed November 10, 2003 and July 3, 2001. They have been placed in the application file and the information referred to therein has been considered by the examiner.

Oath / Declaration

5. The Office acknowledges receipt of a properly signed oath/declaration filed April 30, 2001.

Drawings

6. The drawings are objected to because of minor informalities: hand-written descriptions in FIGs. 2-6 and 8-9 (version 4/30/2001). FIG. 7 already has a replacement sheet filed on 9/20/2004.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement-drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

7. Claim 15 is objected to because of the following informalities: "the JAVA module includings at least". Appropriate correction is required.

Double Patenting

8. From the record, claims 1, 3-4, 6-8, and 10-18 are under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-13 of a co-pending application 09/833,845. Also, claims 19-20 are under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-13 of the co-pending application in view of Anderson, XP-002239737.

Response to Arguments

9. Applicant's arguments with respect to claims 1, 3-4, 6-8, 10-20 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections – 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 1, 3-4, 6-8, and 10-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,920,725 to Ma et al. (art of record, hereinafter "Ma") in view of US Patent No. 6,889,227 to Hamilton (art made of record, hereinafter

Art Unit: 2192

"Hamilton") and further in view of US Patent No. 6,298,478 to Nally et al. (art made of record, hereinafter "Nally").

Claim 1:

Ma discloses *a method for upgrading managed state for a JAVA based application* (e.g., FIG. 3 and related text, column 6, lines 31-67), the method comprising:

generating an upgraded state object (e.g., column 8, lines 55; FIG. 3 and items 68', 68, and related text), wherein the upgraded state object is generated by upgrading a physical schema using data stored in a repository that is part of the databases (e.g., FIG. 3, item 62 and related text, column 4, lines 42-48);

transferring the state stored in the original state object to the upgraded state object (e.g., column 9, lines 20-27; column 11, lines 25-40);

providing state management for the original entity bean using the upgraded state object (e.g., FIG. 8, items 152, 144, and related text).

Ma does not explicitly disclose *executing a JAVA module on a server, wherein the JAVA module is in a middle-tier between a client browser and databases, the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean.*

However, in an analogous art of executing a JAVA module in multi-tier computer environment, Hamilton discloses executing a JAVA module on a server, wherein the

Art Unit: 2192

JAVA module is in a middle-tier between a client browser and databases (e.g., FIG. 1 and related text, column 4, line 48 to column 5, line 52); and

column 1, lines 47-52, "In a three tier environment, a client system (first tier) has a GUI that communicates with an application running on an application server (second tier) which in turn communicates with a database server (third tier) for access to and storage of data").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the teachings of Ma by having said three-tier environment as taught by Hamilton. One would have been motivated because the three-tier environments enable business applications on the second tier to be modified without having to substantially modify each client system as suggested by Hamilton (column 1, lines 52-54).

Ma and Hamilton do not explicitly disclose the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean.

However, in an analogous art of managing Enterprise JavaBeans (EJBs) and maintaining an integrity of the EJBs, Nally discloses said JAVA module is an Enterprise JavaBean (EJB) including an original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean (e.g., column 1, lines 52-58, "For an EJB, the executable business logic is stored within the entity bean. An EJB's business methods are invoked by sending a message to the EJB's wrapper, where a "wrapper" is the Java term for

Art Unit: 2192

functionality required to adapt an EJB to its container, and a "container" is the Java terminology for the run-time environment in which an EJB (including the entity bean) is executed"; and

column 13, lines 14-20, "The EJB specification divides EJBs logically into two parts: an EJB Object, and an entity Bean. FIG. 5 shows how these elements are logically structured, where an EJB 500 is partitioned into an EJB Object 510 and a version 520 of an entity Bean. The EJB Object 510 functions as a wrapper for the EJB"; and

column 13, lines 57-65, "FIG. 5 shows multiple versions of one entity bean conceptually with versions 521 and 522 in addition to version 520. FIG. 5 does not show version status data, business logic, and instance data for versions other than 520; however, it is to be understood that each version of an entity bean has each of these three types of information according to the novel features of the present invention").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Ma-Hamilton-Nally as taught by Nally. One would have been motivated to enhance the system in order to support multiple users or applications that require the capability to access the same EJB at substantially the same time. This mechanism comprises managing multiple concurrent and/or nested transactions against EJBs, and creating and managing versions of EJBs within the transactions as suggested by Nally (column 9, line 48 to column 10, line 21).

Art Unit: 2192

Claim 3:

The rejection of base claim 1 is incorporated. Ma also discloses *comprising the operation of managing the state of the upgraded entity bean using the upgraded state object* as addressed in claim 1 above (e.g., FIG. 8, items 146, 147, 152 and related text, column 11, lines 1-55, "Object adaptor 80 also updates the client class definitions. An invalidate notice is sent to the client's object cache, invalidating the client. However, old client object 144 has already begun to reference new server object 152, which was created based on new classes 140 when old client object 144 instantiated the server object. Since old client object 144 has an old interface while new server object 152 has a new interface, an error occurs. An error is signaled from new server object 152 to old client object 144. An error-handling routine in old client object 144 determines that old client object 144 is invalid. The error-handling routine then re-loads old client object 144, which becomes new client object 147. The state of old client object 144 is transferred to new client object 147 when it is created from new classes 140. New client object 147 can then re-access or re-create new server object 152 and use the new interface").

Claim 4:

The rejection of base claim 3 is incorporated. Ma also discloses *both the original entity bean and the original state object are disabled* (e.g., column 4, lines 59-63, "Other server objects and other client objects continue to run while the object adaptor invalidates the obsolete objects and creates the new server objects and the new client

Art Unit: 2192

objects. Thus the distributed-object client-server application is updated while running"; and

column 5, lines 17-21, "A cache invalidation means in the object adaptor invalidates copies of the obsolete objects by invalidating obsolete class definitions indexed by the client caches. Thus obsolete client objects and classes are invalidated through the client caches").

Claim 6:

The rejection of base claim 4 is incorporated. Ma also discloses *functionality of the JAVA module is not disrupted when the upgraded state object is generated* (e.g., page 1, paragraph [57], "A distributed client-server application is modified while running. The application is not stopped so that updating of objects is transparent. A meta server catalogs all object classes for both the server and the clients. Modifications are specified by a run-time update tool and converted to change commands"; and

column 4, lines 59-63, "Other server objects and other client objects continue to run while the object adaptor invalidates the obsolete objects and creates the new server objects and the new client objects. Thus the distributed-object client-server application is updated while running").

Claim 7:

The rejection of base claim 4 is incorporated. Ma also discloses *functionality of the JAVA application is not disrupted when the JAVA module is upgraded* (e.g., page 1,

Art Unit: 2192

paragraph [57], "A distributed client-server application is modified while running. The application is not stopped so that updating of objects is transparent. A meta server catalogs all object classes for both the server and the clients. Modifications are specified by a run-time update tool and converted to change commands"; and

column 4, lines 59-63, "Other server objects and other client objects continue to run while the object adaptor invalidates the obsolete objects and creates the new server objects and the new client objects. Thus the distributed-object client-server application is updated while running").

Claim 8:

Ma discloses a *JAVA platform capable of performing an online upgrade on a JAVA application, the JAVA platform comprising:*

a repository that is part of the databases and having upgraded class files for the original entity bean and upgraded class files for the original state object (e.g., FIG. 3 item 62 Repository and related text, column 6, lines 39-51);

wherein the original state object is upgraded by generating an upgraded state object using upgraded class files from the repository, and transferring the state stored in the original state object to the upgraded state object (e.g., column 11, lines 25-40); and

an upgrade entity bean is created using data from the repository as the JAVA platform is upgraded (e.g., column 6, lines 19-23; column 7, lines 19-39; and column 9, lines 20-22).

Ma does not explicitly disclose a JAVA module in a middle tier between a client browser and databases, the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, wherein the original state object storing a state of the original entity bean, and wherein the state object provides state management for the original entity bean.

However, in an analogous art of executing a JAVA module in a multi-tier computer environment, Hamilton discloses executing a JAVA module on a server, wherein the JAVA module is in a middle-tier between a client browser and databases (e.g., FIG. 1 and related text, column 4, line 48 to column 5, line 52); and

column 1, lines 47-52, "In a three tier environment, a client system (first tier) has a GUI that communicates with an application running on an application server (second tier) which in turn communicates with a database server (third tier) for access to and storage of data").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the teachings of Ma by having said three-tier environment as taught by Hamilton. One would have been motivated because the three-tier environments enable business applications on the second tier to be modified without having to substantially modify each client system as suggested by Hamilton (column 1, lines 52-54).

Ma and Hamilton do not explicitly disclose the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, wherein the original state object storing a state of the original entity

Art Unit: 2192

bean, and wherein the state object provides state management for the original entity bean.

However, in an analogous art of managing Enterprise JavaBeans (EJBs) and maintaining an integrity of the EJBs, Nally discloses said JAVA module is an Enterprise JavaBean (EJB) including an original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean (e.g., column 1, lines 52-58, "For an EJB, the executable business logic is stored within the entity bean. An EJB's business methods are invoked by sending a message to the EJB's wrapper, where a "wrapper" is the Java term for functionality required to adapt an EJB to its container, and a "container" is the Java terminology for the run-time environment in which an EJB (including the entity bean) is executed"; and

column 13, lines 14-20, "The EJB specification divides EJBs logically into two parts: an EJB Object, and an entity Bean. FIG. 5 shows how these elements are logically structured, where an EJB 500 is partitioned into an EJB Object 510 and a version 520 of an entity Bean. The EJB Object 510 functions as a wrapper for the EJB"; and

column 13, lines 57-65, "FIG. 5 shows multiple versions of one entity bean conceptually with versions 521 and 522 in addition to version 520. FIG. 5 does not show version status data, business logic, and instance data for versions other than 520; however, it is to be understood that each version of an entity bean has each of these three types of information according to the novel features of the present invention").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Ma-Hamilton-Nally as taught by Nally. One would have been motivated to enhance the system in order to support multiple users or applications that require the capability to access the same EJB at substantially the same time. This mechanism comprises managing multiple concurrent and/or nested transactions against EJBs, and creating and managing versions of EJBs within the transactions as suggested by Nally (column 9, line 48 to column 10, line 21).

Claim 10:

The rejection of base claim 8 is incorporated. Ma also discloses *the state of the upgraded entity bean is managed using the upgraded state object*.

Claim 10 recites the same limitations as those of claim 3, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of claim 3, they also teach all of the limitations of claim 10.

Claim 11:

The rejection of bas claim 10 is incorporated. Ma also discloses *both the original entity bean and the original state object are disabled*.

Claim 11 recites the same limitations as those of claim 4, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of claim 4, they also teach all of the limitations of claim 11.

Claim 12:

The rejection of base claim 8 is incorporated. Ma also discloses *the upgraded state object is generated by upgrading a physical schema using data stored in the repository* (e.g., column 5, line 66 to column 6, line 6, "The inventors have realized that central repository techniques can be used for cataloging and storage of program elements. The inventors construct a meta server that stores the object descriptors for all programming objects and uses a database to provide non-volatile storage to this repository of object descriptors. New instances of objects are created from the object description fetched from the meta server's database"; and

column 6, lines 3-51, "Meta server's database 62 is a non-volatile repository of all object class definitions for the distributed application. Even remote objects that are instantiated only on remote clients and not on the server have their class definitions stored persistently in meta server's database 62. The class definitions include the interface definitions, attributes, procedures, and grouping of objects into folders. The class definitions contain the blueprint of objects that are inherited by each instance of an object generated or instantiated. Every time an object is instantiated, the object class definition in meta server 70 serves as the blueprint, although a cache of that object class definition may be used to speed up object creation").

Claim 13:

The rejection of base claim 12 is incorporated. Ma also discloses *functionality of the JAVA module is not disrupted when the JAVA module is upgraded.*

Claim 13 recites the same limitations as those of claim 6, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of claim 6, they also teach all of the limitations of claim 13.

Claim 14:

The rejection of base claim 13 is incorporated. Ma also discloses *functionality of the JAVA application is not disrupted when the JAVA module is upgraded.*

Claim 14 recites the same limitations as those of claim 7, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of claim 7, they also teach all of the limitations of claim 14.

Claims 15-18:

Claims 15-18 recite the same limitations as those of the method claims 1, 3-4, 6-7, and 12, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of claims 1, 3-4, 6-7, and 12, they also teach all of the limitations of claims 15-18.

12. Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ma-Hamilton-Nally as applied to claim 18 above, and further in view of Anderson et al.,

Art Unit: 2192

"Dynamic Code Update in JDrums", XP-002,249,737, published in June 2000 (art of record, hereinafter "Anderson").

Claim 19:

The rejection of base claim 18 is incorporated. Ma, Hamilton, and Nally do not explicitly disclose *the original state object and the upgraded state object are respectively classified into a particular state management unit.*

However, in an analogous art, Anderson discloses the original state object and the upgraded state object are respectively classified (e.g., page 3, left column, section TOOLS, "JDrums provides a tool for automatically generating a conversion skeleton class. which includes all imports, fields and methods that are used to refer to the old and new version of a specific class. A class might have a more or less complex relationship to other classes. The generating tool should in any case figure this out and reflect it in the conversion class.

The tool is connected to the JDrums-enabled JVM through the communication layer and can thereby retrieve information about the old class that is to be updated. Also, it gets information about the enhanced, new class, from a local source.

By automatically generating a conversion class, we minimize the risk of using non-matching fields and methods in our mirror of the old and new class. For instance if the wrong field is used. the compiler will complain")

into a particular state management unit (e.g., pages 2-3, section STRATEGY, "To be able to perform a conversion it is necessary that we have all the information about the component that we intend to update. This is achieved by consulting the running JVM. Alternatively we could have inspected a non-executing entity, e.g. the actual source code. The drawback doing this is that in some cases you might not have access to this entity. We also need information about the updated component. The information gathered so far can be used to generate a skeleton describing attributes of both components. This constitutes, together with the updated component, the conversion package. In the prototype the updates are restricted to the methods and attributes of the class in the conversion package, i.e. members of inherited classes are not accessible, neither is the inheritance graph. The restriction lies not in the JDrums JVM, but in tool which generates the skeleton.

When the conversion package has been created it is sent to the JDrums-enabled JVM through the communication layer. The JVM has been specifically equipped so that it uses the conversion package to reconfigure the running application").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Ma-Hamilton-Nally and Anderson as taught by Anderson. One would have been motivated to enhance the system as taught by Anderson (e.g., page 1, column 1, section Introduction).

Claim 20:

The rejection of base claim 19 is incorporated. Claim 20 recites the limitations, wherein all claimed limitations have been addressed and/or set forth in claim 19 above. Therefore, as the references teach all of the limitations of claim 19, they also teach all of the limitations of claim 20.

Conclusion

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

14. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone is (571) 272 8570. The examiner can normally be reached on Monday – Friday from 6:30AM to 3:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.

Art Unit: 2192

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. Dao



TUAN DAM
SUPERVISORY PATENT EXAMINER